

9. Probabilistic Hough transform [Электронный ресурс] / The university of Edinburgh. — 2017. / Режим доступа: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/AV1011/macdonald.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/macdonald.pdf). — Дата доступа: 25.03.2017.
10. Median filter [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: [https://en.wikipedia.org/wiki/Median\\_filter](https://en.wikipedia.org/wiki/Median_filter). — Дата доступа: 25.03.2017.
11. Gaussian filter [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: [https://en.wikipedia.org/wiki/Gaussian\\_filter](https://en.wikipedia.org/wiki/Gaussian_filter). — Дата доступа: 25.03.2017.
12. Билатеральный фильтр [Электронный ресурс] / Studopedia. — 2017. / Режим доступа: <http://studopedia.info/3-111442.html>. — Дата доступа: 25.03.2017.
13. Ximgproc [Электронный ресурс] / OpenCV. — 2017. / Режим доступа: <http://docs.opencv.org/3.0-beta/modules/ximgproc/doc/ximgproc.html>. — Дата доступа: 25.03.2017.
14. Sauvola binarization [Электронный ресурс] / Mathworks. — 2017. / Режим доступа: <https://www.mathworks.com/matlabcentral/fileexchange/40266-sauvola-local-image-thresholding>. — Дата доступа: 25.03.2017.
15. Grayscale [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: <https://en.wikipedia.org/wiki/Grayscale>. — Дата доступа: 25.03.2017.

УДК 004.451.9

студ. А. Ф. Булова, Е. И. Акшевская

Науч. рук. доц. Н. Н. Буснюк

(кафедра информационных систем и технологии, БГТУ)

**BULL MILKMAN**

## **ИЛИ КАК СОЗДАТЬ ХОРОШИЙ ПРОТОТИП ИГРЫ ЗА 48 ЧАСОВ?**

Главный вопрос: “Почему так быстро?!”. Весь прототип игры собирался в режиме хакатона, который проходил в Минске в марте 2017 г. Основной его задачей было собрать рабочую версию продукта за 48 часов. Так как большинство игр для мобильных платформ используют достаточно простую механику – за основу мы взяли Unity как игровой движок, и добавили качественной векторной графики.

Однако главной целью для нас было пройти полный цикл разработки: от идеи до прототипа.

### Используемые технологии:

- 1) Основы программирования на C#.
- 2) Графический движок Unity.
- 3) Векторных редактор Illustrator.

**Развитие прототипа.** Согласно первоначальному замыслу, игроку предстоит примерить роль быка-молочника, отбиваясь от главных любителей молока – котов, норовящих украсть его груз. Чтобы это сделать необходимо пальцем «тапать», т.е. нажимать на экран, убирая препятствия и обезвреживая врагов. В конце уровня ждет цель путешествия в виде молочного завода, куда необходимо доставить ценный груз. Препятствия исчезают, а враги движутся в обратном направлении, не причиняя вред молоку. Так как геймплей призывает игрока постоянно быть сосредоточенным на устранении препятствий, анимацию (рис. 1) мы постарались сделать плавной, поддерживающей динамичный темп всей игры.

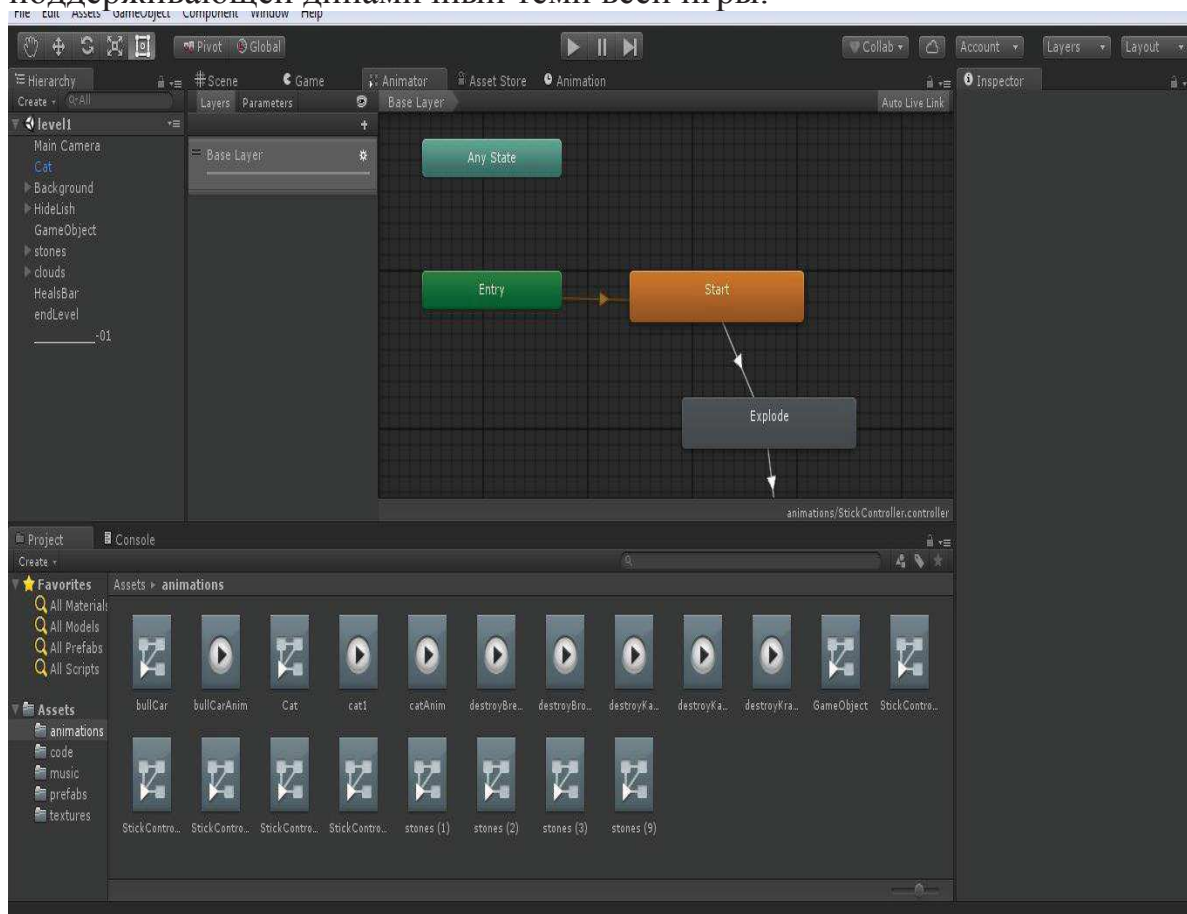


Рисунок 1 (Пример смены анимаций в компоненте «аниматор», Unity)

После того, как игровой процесс был определен, мы приступили к добавлению контента и оформления. Так за следующие пару часов появились: заставка в меню, несколько вариантов локаций, которые сменяли друг друга на разных уровнях, другие декорации, такие как кусты, деревья, горы, а также несколько видов врагов.

**Графика и анимация.** Все персонажи претерпевали изменения по сравнению с первоначальными образами, добавлялось все больше деталей. (рис. 2)

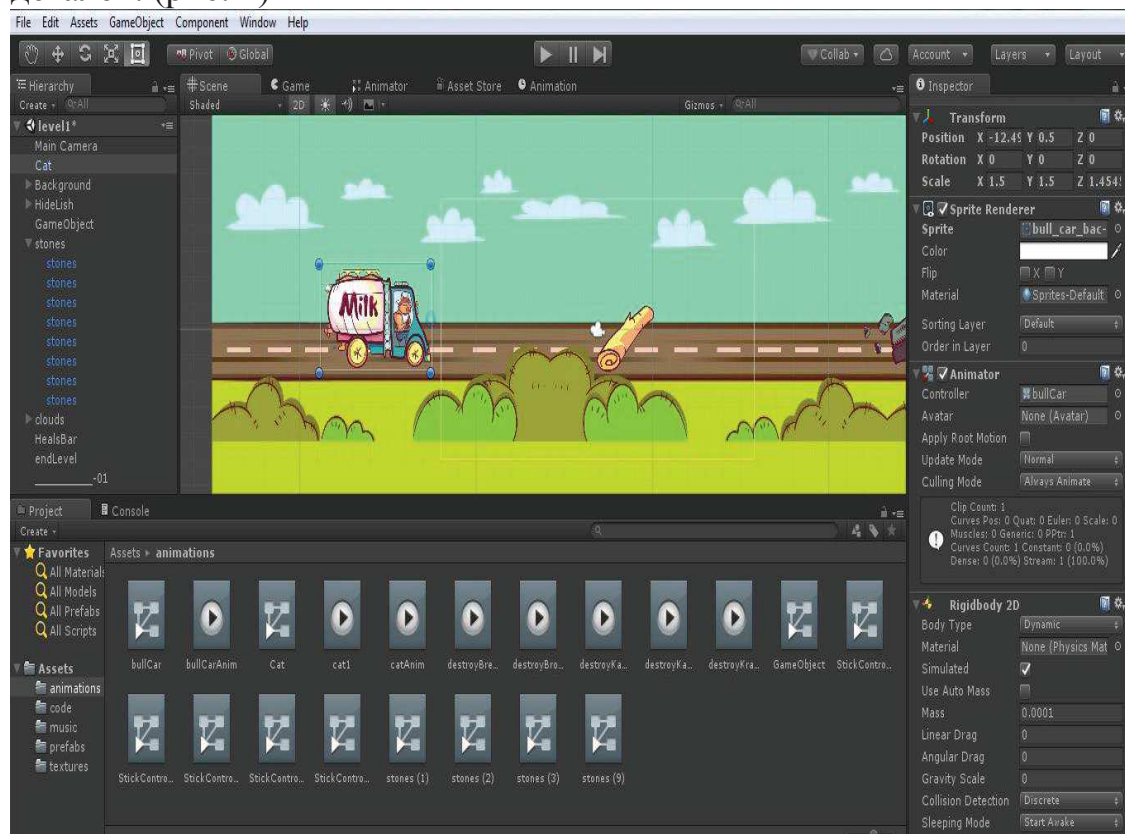


Рисунок 2 (Скриншот рабочей сцены из игры)

**Оптимизация.** Когда все было почти готово, несмотря на тесты на реальных девайсах, мы столкнулись с проблемой производительности, оптимизация заняла ощутимое количество времени. У нас были неполадки с пересечением коллайдеров, при которых персонаж менял траекторию пути, а также съедал две жизни, если враг успевал пересечь границу коллайдера главного персонажа, переходы анимаций по определенному событию (функции) в скрипте, а также мелкие баги, которые нам удалось оперативно исправить.

Для врагов мы также применили “запекание” физики в анимацию. На каждого врага было записано по 2 анимации. На деле выглядит почти так же, как и с применением симуляции физики.

**Заключение.** Таков был наш первый опыт разработки в подобном формате. (рис. 3, рис. 4) Это дало нам возможность погрузиться в рабочую атмосферу, где важно время и качество итогового продукта, а также дало неплохой опыт по поиску решений возникнувших задач и проблем, либо поиску обходных путей, что, по нашему мнению, важно в работе любого программиста. В свою очередь конференция в БГТУ позволила представить нашу работу на суд зрителей, в том числе преподавателей, и получить дополнительную оценку специалистов и опыт представления своих программ, а также толчок к развитию в этом направлении, улучшению навыков программистов и художников, и созданию в будущем новых игр.



Рисунок 3 (Скриншот готового прототипа игры. (начальный экран))



Рисунок 4 (Скриншот готового прототипа игры. (локация 1 уровня))

УДК 519.62 + 531.3 + 521.1

Студ. П. С. Шенец

Науч. рук. ст. преподаватель А. С. Наркевич  
(кафедра информационных систем и технологий, БГТУ)

## МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛ ПОД ДЕЙСТВИЕМ ФОНА ГРАВИТОНОВ

В 1930-х годах было обнаружено, что в скоплениях галактик отдельные галактики движутся заметно быстрее, чем это возможно, если их удерживает вместе притяжение светящейся массы скопления. Ф. Цвикки показал, что отношение необходимой массы к видимой составляет около 500 [1], т.е. большая часть массы не видна. Позже в спиральных галактиках наблюдения обнаружили слишком высокие скорости звезд на их периферии, причем эти скорости стремятся к постоянному пределу, зависящему от полной светимости галактики. Проблему недостающей массы (так ее называли) можно решить, предположив, что существует некая темная материя, заполняющая галактики и их скопления [2]. Это принятая на сегодня гипотеза. Другой вариант – это возможная модификация уравнения движения